In the Claims:

Please cancel claims 39 and 50, as indicated in the listing below.

1. (Previously presented) A system, comprising:

a distributed store configured to store a primary state of session data configured for access by a plurality of application servers, wherein the session data comprises one or more attributes;

a first application server of the plurality of application servers coupled to the distributed store, and configured to store a client state of the session data comprising one or more attributes, wherein the first application server is configured to provide processes executing within the first application server with access to the client state of the session data;

wherein the first application server is configured to:

track mutable accesses of the attributes in the client state of the session data;

perform an object graph comparison of mutably accessed attributes of the client state of the session data with a benchmark version of the client state of the session data to determine a subset of modified attributes, wherein the benchmark version of the client state of the session data comprises a previous version of the attributes in the client state of the session data; and

wherein the distributed store is configured to synchronize the primary state of the session data with the client state of the session data according to the subset of modified attributes.

2.   (Previously presented)   The system as recited in claim 1, wherein a the mutable accesses comprises a write access to an attribute of the client state of the session data.

3.   (Previously presented)   The system as recited in claim 1, wherein, to synchronize the primary state of the session data with the client state of the session data, the distributed store is further configured to update the primary state of the session data using the subset of modified attributes.

4.   (Previously presented)   The system as recited in claim 1, wherein the distributed store is further configured to lock the primary state of the session data for access by a process executing on the first application server, wherein, while the primary state of the session data is locked for the process, other processes on the first application server cannot access the primary state of the session data.

5.   (Previously presented)   The system as recited in claim 4, wherein the distributed store is further configured to lock the primary state of the session data for access by a process executing on the first application server, wherein, while the primary state of the session data is locked for the process, other processes on others of the plurality of application servers cannot access the primary state of the session data.

6.   (Previously presented)   The system as recited in claim 4, wherein the distributed store is further configured to request the process to release the lock on the primary state of the session data, wherein the process is configured to release the lock on the primary state of the session data in response to said request.

7. (Previously presented) The system as recited in claim 4, wherein the process is configured to release the lock when locked access to the primary state of the session data is no longer required by the process.

8. (Previously presented) The system as recited in claim 4, wherein the distributed store is further configured to lock at least a portion of the primary state of the session data for access by a thread executing within the process, wherein, while the at least a portion of the primary state of the session data is locked for the thread, other threads executing within the process cannot access the at least a portion of the primary state of the session data.

9. (Previously presented) The system as recited in claim 1, wherein the primary state of the session data is distributed across a plurality of devices.

10. (Previously presented) The system as recited in claim 1 configured to store a plurality of instances of the primary state of the session data, wherein the system is configured to:

> determine differences between a first instance of the primary state of the session data and a benchmark instance of the primary state of the session data; and

> synchronize a second instance of the primary state of the session data with the first instance of the primary state of the session data according to the determined differences.

11. (Previously presented) The system as recited in claim 10, wherein to determine differences between a first instance of the primary state of the session data and a benchmark instance of the primary state of the session data, the system is further configured to perform a binary comparison of the first instance of the primary state and the benchmark instance of the primary state.

12. (Previously presented) The system as recited in claim 10, wherein to determine differences between a first instance of the primary state of the session data and a benchmark instance of the primary state of the session data, the system is further configured to perform an object graph comparison of the first instance of the primary

state and the benchmark instance of the primary state.

13. (Previously presented) The system as recited in claim 1, wherein the first application server is further configured to:

track accesses of mutable attributes of the client state of the session data;

perform an object graph comparison of accessed mutable attributes and a benchmark version of the client state of the session data to determine a subset of modified mutable attributes; and

wherein, to synchronize the primary state of the session data with the client state of the session data, the distributed store is further configured to use the subset of modified mutable attributes.

14. (Previously presented) A system, comprising:

a plurality of application servers, wherein each of the plurality of application servers comprises a client state of session data comprising one or more attributes, wherein each of the application servers is configured to provide access to the corresponding client state of session data to processes executing within the particular application server;

a distributed store coupled to the plurality of application servers, comprising a primary state of the session data configured for access by the plurality of application servers;

wherein each of the plurality of application servers is configured to:

track mutable accesses of the attributes in the client state of session data of the particular application server;

perform an object graph comparison of mutably accessed attributes with a benchmark version of the client state of session data to determine a subset of modified attributes, wherein the benchmark version of the client state of session data comprises a previous version of the attributes in the client state of session data;

wherein the distributed store is configured to synchronize the primary state of session data with the client state of session data according to the subset of modified attributes.

15. (Previously presented) The system as recited in claim 14, wherein a the mutable accesses comprises a write access to an attribute of the client state of the session data.

16. (Previously presented) The system as recited in claim 14, wherein, to synchronize the primary state with the client state of the session data, the distributed store is further configured to update the primary state of the session data with the subset of modified attributes.

17. (Previously presented) The system as recited in claim 14, wherein the distributed store is further configured to provide locked access to the primary state of the session data to processes executing on the plurality of application servers, wherein, while the primary state of the session data is locked for a process, other processes cannot access the primary state of the session data.

18. (Previously presented) The system as recited in claim 17, wherein the distributed store is further configured to request the process having the lock to release the lock, wherein the process having the lock is configured to release the lock in response to said request.

19. (Previously presented) The system as recited in claim 17, wherein the process having the lock is configured to release the lock when locked access to the primary state is no longer required by the process having the lock.

20. (Previously presented) The system as recited in claim 17, wherein the distributed store is further configured to lock at least a portion of the primary state of the session data for access by a thread executing within the process, wherein, while the at least a portion of the primary state of the session data is locked for the thread, other threads executing within the process cannot access the at least a portion of the primary state of the session data.

21. (Previously presented) The system as recited in claim 14, wherein the primary state of the session data is distributed across a plurality of devices.

22. (Previously presented) The system as recited in claim 14, configured to store a plurality of instances of the primary state of the session data, wherein the system is configured to:

> determine differences between a first instance of the primary state of the session data and a benchmark instance of the primary state of the session data; and

> synchronize a second instance of the primary state of the session data with the first instance of the primary state of the session data according to the determined differences.

23. (Previously presented) The system as recited in claim 22, wherein to determine differences between a first instance of the primary state of the session data and a benchmark instance of the primary state of the session data, the system is further configured to perform a binary comparison of the first instance of the primary state and the benchmark instance of the primary state.

24. (Previously presented) The system as recited in claim 22, wherein to determine differences between a first instance of the primary state of the session data and a benchmark instance of the primary state of the session data, the system is further configured to perform an object graph comparison of the first instance of the primary state and the benchmark instance of the primary state.

25. (Previously presented) The system as recited in claim 14, wherein each of the plurality of application servers is further configured to:

track accesses of mutable attributes in the client state of the session data;

perform an object graph comparison of the accessed mutable attributes and the benchmark version of the client state of the session data to determine a subset of modified mutable attributes; and

wherein, to synchronize the primary state of the session data with the client state of the session data, the distributed store is further configured to use the subset of modified mutable attributes.

26. (Previously presented) A system, comprising:

means to lock access to a primary state of session data configured for access by a plurality of application servers for a process executing on one of the plurality of application servers, wherein the session data comprises a plurality of attributes;

wherein, while the primary state of the session data is locked for the process, other processes cannot access the primary state of the session data;

wherein each of the plurality of application servers comprises a client state of the session data accessible to processes executing within the application

server;

means for each of the application servers to:

track accesses of mutable attributes in the client state of the session data;

determine a set of mutably accessed attributes of the client state of the session data of the particular application server;

determine a subset of the set of mutably accessed attributes that are modified in respect to the primary state of the session data; and

means to synchronize the primary state of the session data with the client state of the session data using the subset of modified attributes.

27.   (Previously presented)   The system as recited in claim 26, wherein said means for each of the application servers to determine a set of mutably accessed attributes of the client state of the session data of the particular application server comprises means for tracking mutable accesses of the attributes in the client state of the session data.

28.   (Previously presented)   The system as recited in claim 26, wherein said means for each of the application servers to determine a subset of the set of mutably accessed attributes that are modified in respect to the primary state of the session data comprises means for performing an object graph comparison of the mutably accessed attributes with a benchmark version of the client state of the session data to determine the subset of modified attributes, wherein the benchmark version of the client state of the session data comprises a previous version of the attributes in the client state of the session data.

29. (Previously presented) A computer implemented method, comprising:

tracking mutable accesses of a plurality of attributes of a client state of session data, wherein the client state of the session data is associated with an application server;

performing an object graph comparison of mutably accessed attributes with a benchmark version of the client state of the session data to determine a subset of modified attributes, wherein the benchmark version of the client state of the session data comprises a previous version of the attributes in the client state of the session data; and

synchronizing a primary state of the session data with the client state of the session data according to the subset of modified attributes, wherein the primary state of the session data is accessible by a plurality of application servers.

30. (Previously presented) The method as recited in claim 29, wherein a the mutable accesses comprises a write access to an attribute.

31. (Previously presented) The method as recited in claim 29, wherein said synchronizing comprises updating the primary state of the session data using the subset of modified attributes.

32. (Previously presented) The method as recited in claim 29, further comprising locking the primary state of the session data for access by a process executing on the application server, wherein, while the primary state of the session data is locked for access by the process, other processes on the plurality of application servers cannot access the primary state of the session data.

33. (Previously presented) The method as recited in claim 32, further comprising:

the process receiving a request to release locked access to the primary state of the session data; and

the process releasing the locked access to the primary state of the session data in response to said request.

34. (Previously presented) The method as recited in claim 32, further comprising releasing locked access to the primary state of the session data when no longer required by the process.

35. (Previously presented) The method as recited in claim 32, further comprising locking one or more attributes of the primary state of the session data for access by a thread executing within the process, wherein, while the one or more attributes are locked for access by the thread, other threads executing within the process cannot access the one or more attributes.

36. (Previously presented) The method as recited in claim 29, further comprising:

determining differences between the primary state of the session data and a benchmark version of the primary state of the session data; and

synchronizing another instance of the primary state of the session data with the primary state of the session data using the determined differences.

37. (Previously presented) The method as recited in claim 36, wherein said determining differences between the primary state of the session data and a benchmark version of the primary state of the session data comprises performing a binary comparison of the primary state of the session data and the benchmark version of the primary state of the session data.

38. (Previously presented) The method as recited in claim 36, wherein said determining differences between the primary state of the session data and a benchmark version of the primary state of the session data comprises performing an object graph comparison of the primary state of the session data and the benchmark version of the primary state of the session data.

39. (Canceled)

40. (Previously presented) A tangible, computer accessible medium, comprising software instructions executable to implement:

> tracking mutable accesses of a plurality of attributes of a client state of session data, wherein the client state of the session data is associated with an application server;

> performing an object graph comparison of mutably accessed attributes with a benchmark version of the client state of the session data to determine a subset of modified attributes, wherein the benchmark version of the client state of the session data comprises a previous version of the attributes in the client state of the session data; and

> synchronizing a primary state of the session data with the client state of the session data using the subset of modified attributes, wherein the primary state of the session data is accessible by a plurality of application servers.

41. (Previously presented) The tangible, computer accessible medium as recited in claim 40, wherein a the mutable accesses comprises a write access to an attribute.

42. (Previously presented) The tangible, computer accessible medium as recited in claim 40, wherein, in said synchronizing, the software instructions are further

executable to implement updating the primary state of the session data using the subset of modified attributes.

43. (Previously presented) The tangible, computer accessible medium as recited in claim 40, wherein the software instructions are further executable to implement locking the primary state of the session data for access by a process executing on the application server, wherein, while the primary state of the session data is locked for access by the process, other processes on the plurality of application servers cannot access the primary state of the session data.

44. (Previously presented) The tangible, computer accessible medium as recited in claim 43, wherein the software instructions are further executable to implement:

the process receiving a request to release locked access to the primary state of the session data; and

the process releasing the locked access to the primary state of the session data in response to said request.

45. (Previously presented) The tangible, computer accessible medium as recited in claim 43, wherein the software instructions are further executable to implement releasing locked access to the primary state of the session data when no longer required by the process.

46. (Previously presented) The tangible, computer accessible medium as recited in claim 43, wherein the software instructions are further executable to implement locking one or more attributes of the primary state of the session data for access by a thread executing within the process, wherein, while the one or more attributes are locked for access by the thread, other threads executing within the process cannot access the one or more attributes.

47. (Previously presented) The tangible, computer accessible medium as recited in claim 40, wherein the software instructions are further executable to implement:

determining differences between the primary state of the session data and a benchmark version of the primary state of the session data; and

synchronizing another instance of the primary state of the session data with the primary state of the session data using the determined differences.

48. (Previously presented) The tangible, computer accessible medium as recited in claim 47, wherein, in determining differences between the primary state of the session data and a benchmark version of the primary state of the session data, the software instructions are further executable to implement a binary comparison of the primary state of the session data and the benchmark version of the primary state of the session data.

49. (Previously presented) The tangible, computer accessible medium as recited in claim 47, wherein, in determining differences between the primary state of the session data and a benchmark version of the primary state of the session data, the software instructions are further executable to implement an object graph comparison of the primary state of the session data and the benchmark version of the primary state of the session data.

50. (Canceled)